Gene Selection from Parkinson's Disease Dataset Using k-Nearest-Neighbor and Differential Evolution Algorithm(DE)

A Thesis
submitted in partial fulfillment of the requirement for the Degree of
Bachelor of Technology in Computer Science & Engineering

Maulana Abul Kalam Azad University of Technology ,West Bengal
May-2023

By

Onkar Halder
Registration No: 030951 of 2019-23
Examination Roll No: 10000119009

Under the Guidance of

Dr. Sriyankar Acharyya
Professor,
Department of Computer Science & Engineering,
Maulana Abul Kalam Azad University of Technology,
Kolkata-741249, W.B., India

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

Maulana Abul Kalam Azad University of Technology, West Bengal

Faculty of Maulana Abul Kalam Azad University of Technology,West Bengal

CERTIFICATE

This is to certify that the dissertation entitled "Gene Selection from Parkinson's Disease Dataset Using k-Nearest-Neighbor and Differential Evolution Algorithm(DE)" has been carried out by Onkar Halder(University Registration No: 030951 of 2019-23) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Bachelor of Technology in Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree to any other University or Institute.

Dr. Sriyankar Acharyya
Professor
Dept. of Computer Science & Engineering
Maulana Abul Kalam Azad University of Technology, W.B.

Dr. Pradyut Sarkar
Head of the Department
Dept. of Computer Science & Engineering
Maulana Abul Kalam Azad University of Technology, W.B.

Faculty of Maulana Abul Kalam Azad University of Technology ,West Bengal

CERTIFICATE OF APPROVAL*

The forgoing thesis is hereby approved as a creditable study of an engineering subject and presented in a manner satisfactory to warrant acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for which it is submitted.

Committee on final examination for the evaluation of the thesis

---

Signature of the Examiner

Signature of the Supervisor

*Only in the case the thesis is approved

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC THESIS

I hereby declare that this thesis entitled "Gene Selection from Parkinson's Disease Dataset Using k-Nearest-Neighbor and Differential Evolution Algorithm(DE)" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Bachelor of Technology in Computer Science and Engineering.

All information has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Onkar Halder

Examination Roll No.: 10000119009

Thesis Title: Gene Selection from Parkinson's Disease Dataset Using k-Nearest-Neighbor and Differential Evolution Algorithm(DE)

Signature of the candidate

ACKNOWLEDGEMENTS

First and foremost, I would like to express my earnest gratitude and heartfelt indebtedness to my advisor, Dr. Sriyankar Acharyya, Department of Computer Science & Engineering, for the privilege and the pleasure, of allowing me to work under his towards my Degree of Bachelor of Technology in Computer Science and Engineering. This work would not have been materialized, but for his whole-hearted help and support. Working under his has been a great experience. I sincerely thank my supervisor, particularly for all the faith he had in me. I am thankful to Prof. Pradyut Sarkar who have acted as Head of the Department of Computer Science & Engineering during the tenure of my studentship. I would also like to show my gratitude to the respected professors of the Department of Computer Science & Engineering for their constant guidance and valuable advices.

Date:

Place:

Onkar Halder
Examination Roll No.: 10000119009
Maulana Abul Kalam Azad University of Technology,West Bengal

Abstract

Parkinson's disease (PD) is a multifaceted neurodegenerative disorder characterized by the progressive deterioration of dopaminergic neurons. The advent of high-throughput genomic technologies has made it possible to access extensive gene expression datasets for investigating the molecular intricacies associated with PD. Nevertheless, the identification of pertinent genes from these datasets remains an arduous task due to the data inherent high dimensionality and noise.

To address this challenge, Propose an innovative approach firstly used DE on benchmark functions afterward optimized the DE,then gene selection in Parkinson's disease datasets by combining the k-Nearest-Neighbor (k-NN) algorithm with the optimized Differential Evolution Algorithm (DE). The k-NN algorithm is employed to identify genes with high relevance to the disease based on their informative nature. Subsequently, the DE algorithm is utilized to refine the gene selection by optimizing the feature subset according to its classification performance.

To evaluate the efficacy of our proposed approach, conducted experiments has been done using a publicly available dataset of gene expression profiles from PD patients and healthy individuals

The experimental results demonstrate that the used k-NN and DE-based gene selection approach surpasses the performance of competing methods in terms of both classification accuracy and the number of selected genes. The selected subset of genes comprises biologically significant candidates that are already associated with PD, thereby validating the effectiveness of our approach.Some of the gene are matched with other researchers reported gene ids.Further it can be implement the same with other type of DE and other algorithm to identify more genes.

Keywords: Gene Selection,Benchmark function,Differential Evolution(DE),kNN,Sample Classification, Gene Expression.

Content

1.Introduction

Machine learning (ML) operates with the automated learning of machines through training and testing without manually programmed. It performs predictions on the basis of data and has several important applications in the field of bioinformatics.[22] Bioinformatics is the term used for the processing of biological data using several approaches based on computation and mathematics. In these modern days, the biological data has grown exponentially, leading to two major problems. One problem is of efficient information storage and the second problem deals with how to mine useful insights from the data. In the field of Bioinformatics, detection of critical disease genes by studying gene expression datasets is crutial.

For conducting further studies and close studying of the genes for medical uses, we could develop a powerful algorithm which would predict the similarities between the genes. Genetic selection is a powerful method for classifying pure samples of differentiated types of genes [21]. In order to develop this strategy, a gene that is expressed specifically in the microarray must be identified. On the other side, a population of disease critical genes are limited.

The final data obtained after processing is in the form of a matrix. The columns (often termed as samples) represents various biological conditions in which genes are tested and rows represents gene indices. Each gene matrix element is a decimal value, after going through primary normalization. In various execution of this algorithm code, the best candidate solutions are taken into account and according to their mean, classification accuracy is calculated.

Gene selection is more of a conceptual framework to define a group of genes that share a common phenomenon. This common phenomenon could be Function, Pathway, Co-localization / Physical location, Co-expression. Protein–protein interactions (PPIs) or physical contacts of high specific establishment between two or more protein molecules [21].

Parkinson's disease affects the portions of the body that are regulated by the nerves and is a progressive central nervous system disorder. [3]. It is difficult to identify the biological cause or detect any abnormal presence of biological activities for Parkinson's disease. In the below section, we discussed how we are approaching the problem to identify the biomarkers from Parkinson's disease.

To research, a large number of genes is difficult using the traditional method, for this work DNA microarray comes in help. M*icroarray* is a laboratory high-throughput technology used to detect the expression for a large number of genes in a single reaction quickly in an efficient manner, DNA microarrays are microscopic slides that have thousands of small spots, and each spot contains a known gene or DNA sequence[1].

Microarray dataset contains microarray gene expression data, it stores the measurement of data, searches through the indexes, and makes it available for other analysis. A classification task is to separate patients based on their gene expression, this type of data is also used to gather information from cells and tissues regarding gene expression differences which will help with disease diagnosis[2].

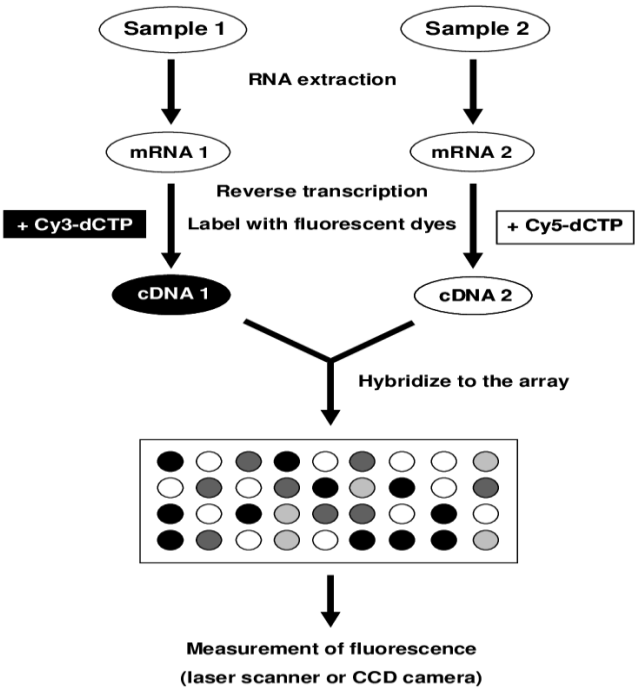


Fig 1.1- Outline of Gene Expression Analysis[11]

To identify critical genes we are using the Differential Evolution Algorithm which is a part of Metaheuristic algorithms. Metaheuristic algorithms are computational intelligence paradigms especially used for solving optimization problems. During the optimization process, these metaheuristic search algorithms can conduct searches with varying levels of exploration and exploitation strengths to find the global or nearly global optimum solution. [4].

- *E*xploration includes the process of finding various solutions in the search space, it can support population variety.
- *Exploitation* refines the information discovered by the search process within the best solutions.

Metaheuristic Search Algorithms can be divided into some categories depending on generating optimal solutions- Evolutionary Algorithms, Swarm Intelligence Algorithms, Human-based Algorithms, and Physics-based Algorithms [4].

Differential Evolution *is* proposed by Storn et al. DE is a population-based Evolutionary Algorithm inspired by Darwin's evolution theory and the 'survival of the fittest' concept. It is one of the most popular optimization algorithms to solve complex optimization problems.

In the below, we discussed the problem described in section 2, section 3.methodology which discusses the method we used, and experimental results in section 4.

1.1. Related Works on Differential Evolution Algorithm

Several researchers had gave their effort to the research areas of DE since 1997 [14][15][16][17][18][19][20].In the year 2009 Neri and Tirronen published their survey on Various modified DE [16].Till 2011 and 2016 ,Das and Suganthan [14] by Das et al [17] rigorously survey the advances of DE. The main topic of this review article is the changes made in DE options available to solve different optimization environments and engineering applications.Jebaraj et al published his review paper upto 2016 [18],it focused on the application of DE to solve economic problems or static and dynamic emissions.Bilal et al [15] explored the recent advances of DE variants upto 2018.Opara and Arabs published their DE works on convergence characteristic,computational complexityans dynamics models of DE [19].

2. Problem Description

Symptoms of Parkinson's disease usually begin gradually and increase over time. People may struggle to walk and speak throughout this stage. This disease also causes mental and behavioral changes, sleep problems, depression, and memory difficulties. The most prominent signs of this disease occur when nerve cells in the basal ganglia become damaged or die. Normally, these nerve cells produce an important brain chemical (dopamine)[3]. Less dopamine is produced when a neuron dies, which results in difficulties speaking or moving.

Parkinson's has some prominent symptoms:

- Tremor in hands, legs, and jaw
- Muscle stiffness
- Impaired balance and coordination.

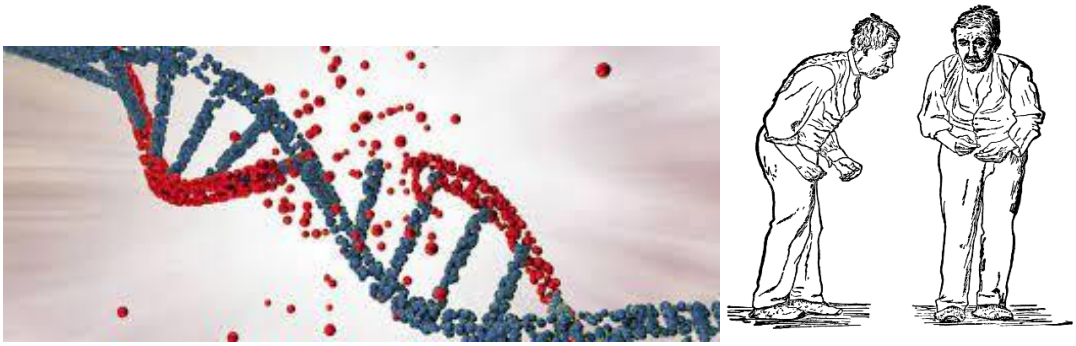
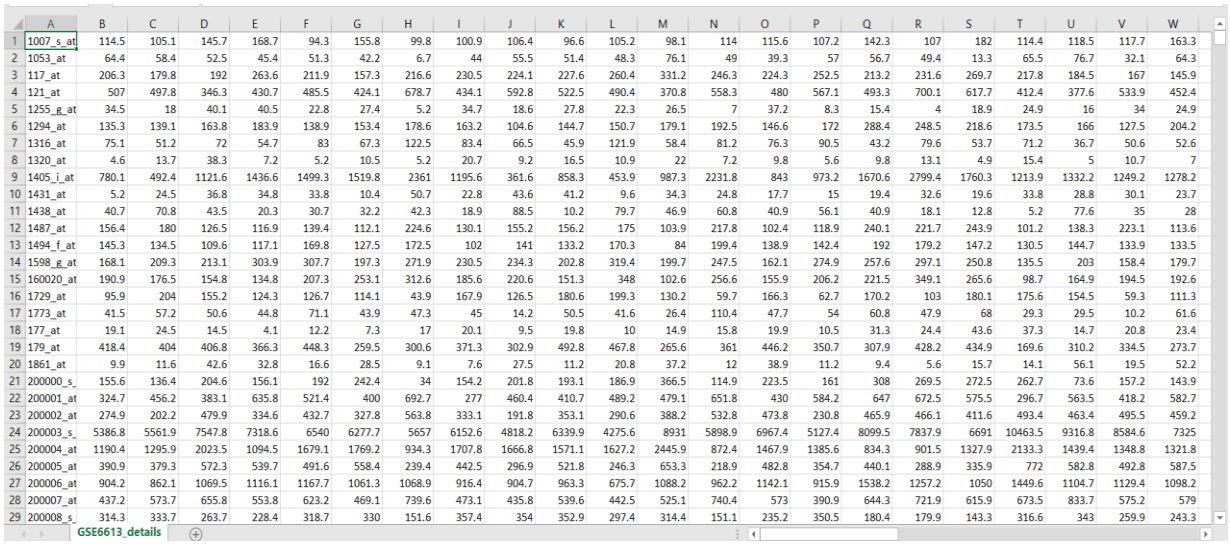


Fig 1.2-Genetic disorder[12][13]

Microarray gene expression data from microarray dataset, which contains the measurement of data is used to identify critical gene for the disease. this dataset is also used to gather

information from cells and tissues regarding gene expression differences which will help with disease diagnosis.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	1007_s_at	114.5	105.1	145.7	168.7	94.3	155.8	99.8	100.9	106.4	96.6	105.2	98.1	114	115.6	107.2	142.3	107	182	114.4	118.5	117.7	163.3
2	1053_at	64.4	58.4	52.5	45.4	51.3	42.2	6.7	44	55.5	51.4	48.3	76.1	49	39.3	57	56.7	49.4	13.3	65.5	76.7	32.1	64.3
3	117_at	206.3	179.8	192	263.6	211.9	157.3	216.6	230.5	224.1	227.6	260.4	331.2	246.3	224.3	252.5	213.2	231.6	269.7	217.8	184.5	167	145.9
4	121_at	507	497.8	346.3	430.7	485.5	424.1	678.7	434.1	592.8	522.5	490.4	370.8	558.3	480	567.1	493.3	700.1	617.7	412.4	377.6	533.9	452.4
5	1255_g_at	34.5	18	40.1	40.5	22.8	27.4	5.2	34.7	18.6	27.8	22.3	26.5	7	37.2	8.3	15.4	4	18.9	24.9	16	34	24.9
6	1294_at	135.3	139.1	163.8	183.9	138.9	153.4	178.6	163.2	104.6	144.7	150.7	179.1	192.5	146.6	172	288.4	248.5	218.6	173.5	166	127.5	204.2
7	1316_at	75.1	51.2	72	54.7	83	67.3	122.5	83.4	66.5	45.9	121.9	58.4	81.2	76.3	90.5	43.2	79.6	53.7	71.2	36.7	50.6	52.6
8	1320_at	4.6	13.7	38.3	7.2	5.2	10.5	5.2	20.7	9.2	16.5	10.9	22	7.2	9.8	5.6	9.8	13.1	4.9	15.4	5	10.7	7
9	1405_i_at	780.1	492.4	1121.6	1436.6	1499.3	1519.8	2361	1195.6	361.6	858.3	453.9	987.3	2231.8	843	973.2	1670.6	2799.4	1760.3	1213.9	1332.2	1249.2	1278.2
10	1431_at	5.2	24.5	36.8	34.8	33.8	10.4	50.7	22.8	43.6	41.2	9.6	34.3	24.8	17.7	15	19.4	32.6	19.6	33.8	28.8	30.1	23.7
11	1438_at	40.7	70.8	43.5	20.3	30.7	32.2	42.3	18.9	88.5	10.2	79.7	46.9	60.8	40.9	56.1	40.9	18.1	12.8	5.2	77.6	35	28
12	1487_at	156.4	180	126.5	116.9	139.4	112.1	224.6	130.1	155.2	156.2	175	103.9	217.8	102.4	118.9	240.1	221.7	243.9	101.2	138.3	223.1	113.6
13	1494_f_at	145.3	134.5	109.6	117.1	169.8	127.5	172.5	102	141	133.2	170.3	84	199.4	138.9	142.4	192	179.2	147.2	130.5	144.7	133.9	133.5
14	1598_g_at	168.1	209.3	213.1	303.9	307.7	197.3	271.9	230.5	234.3	202.8	319.4	199.7	247.5	162.1	274.9	257.6	297.1	250.8	135.5	203	158.4	179.7
15	160020_at	190.9	176.5	154.8	134.8	207.3	253.1	312.6	185.6	220.6	151.3	348	102.6	256.6	155.9	206.2	221.5	349.1	265.6	98.7	164.9	194.5	192.6
16	1729_at	95.9	204	155.2	124.3	126.7	114.1	43.9	167.9	126.5	180.6	199.3	130.2	59.7	166.3	62.7	170.2	103	180.1	175.6	154.5	59.3	111.3
17	1773_at	41.5	57.2	50.6	44.8	71.1	43.9	47.3	45	14.2	50.5	41.6	26.4	110.4	47.7	54	60.8	47.9	68	29.3	29.5	10.2	61.6
18	177_at	19.1	24.5	14.5	4.1	12.2	7.3	17	20.1	9.5	19.8	10	14.9	15.8	19.9	10.5	31.3	24.4	43.6	37.3	14.7	20.8	23.4
19	179_at	418.4	404	406.8	366.3	448.3	259.5	300.6	371.3	302.9	492.8	467.8	265.6	361	446.2	350.7	307.9	428.2	434.9	169.6	310.2	334.5	273.7
20	1861_at	9.9	11.6	42.6	32.8	16.6	28.5	9.1	7.6	27.5	11.2	20.8	37.2	12	38.9	11.2	9.4	5.6	15.7	14.1	56.1	19.5	52.2
21	200000_s_	155.6	136.4	204.6	156.1	192	242.4	34	154.2	201.8	193.1	186.9	366.5	114.9	223.5	161	308	269.5	272.5	262.7	73.6	157.2	143.9
22	200001_at	324.7	456.2	383.1	635.8	521.4	400	692.7	277	460.4	410.7	489.2	479.1	651.8	430	584.2	647	672.5	575.5	296.7	563.5	418.2	582.7
23	200002_at	274.9	202.2	479.9	334.6	432.7	327.8	563.8	333.1	191.8	353.1	290.6	388.2	532.8	473.8	230.8	465.9	466.1	411.6	493.4	463.4	495.5	459.2
24	200003_s_	5386.8	5561.9	7547.8	7318.6	6540	6277.7	5657	6152.6	4818.2	6339.9	4275.6	8931	5898.9	6967.4	5127.4	8099.5	7837.9	6691	10463.5	9316.8	8584.6	7325
25	200004_at	1190.4	1295.9	2023.5	1094.5	1679.1	1769.2	934.3	1707.8	1666.8	1571.1	1627.2	2445.9	872.4	1467.9	1385.6	834.3	901.5	1327.9	2133.3	1439.4	1348.8	1321.8
26	200005_at	390.9	379.3	572.3	539.7	491.6	558.4	239.4	442.5	296.9	521.8	246.3	653.3	218.9	482.8	354.7	440.1	288.9	335.9	772	582.8	492.8	587.5
27	200006_at	904.2	862.1	1069.5	1116.1	1167.7	1061.3	1068.9	916.4	904.7	963.3	675.7	1088.2	962.2	1142.1	915.9	1538.2	1257.2	1050	1449.6	1104.7	1129.4	1098.2
28	200007_at	437.2	573.7	655.8	553.8	623.2	469.1	739.6	473.1	435.8	539.6	442.5	525.1	740.4	573	390.9	644.3	721.9	615.9	673.5	833.7	575.2	579
29	200008_s_	314.3	333.7	263.7	228.4	318.7	330	151.6	357.4	354	352.9	297.4	314.4	151.1	235.2	350.5	180.4	179.9	143.3	316.6	343	259.9	243.3

GSE6613_details

Fig 1.3- Microarray Gene Expression Dataset[10]

A microarray gene expression dataset takes as input. The main motive of the work is to identify a subset which containing *n* genes, such that it can have the most optimal solution. For this classification, ${}^{n}_{d}C$ Possible searches, where *n* is the number of genes in the dataset and *d* is the no of genes in the subset. So, DE is executed to classify the group of genes in these huge gene datsets [21].

2.1 Taking Dataset *M* as Input

The matrix of gene expression dataset *M*, is shown in the table. The columns contains normal samples ($N_j\ stands\ for\ jth\ sample$) and diseased samples ($D_k\ stands\ for\ kth\ sample$). Here, $i \in \{1, 2, \dots\dots, n\}$, $j \in \{1, 2, \dots., maxN\}$ and $k \in \{1, 2, \dots., maxD\}$, where *n* is the number of genes, *maxD* is the number of diseases affected samples in dataset and *maxN* is the number of normal sample.

Gene		*Normal Samples*				*Diseases Samples*			
		N_1	N_2	…	N_{maxN}	D_1	D_2	….	D_{maxD}
G_1		$g^N_{1,1}$	$g^N_{1,2}$	…	$g^N_{1,maxN}$	$g^D_{1,1}$	$g^D_{1,2}$	…	$g^D_{1,maxN}$
G_2		$g^N_{1,1}$	$g^N_{2,2}$	…	$g^N_{2,maxN}$	$g^D_{1,1}$	$g^D_{2,2}$	…	$g^D_{2,maxN}$
G_3		$g^N_{1,1}$	$g^N_{3,2}$	…	$g^N_{3,maxN}$	$g^D_{1,1}$	$g^D_{3,2}$	…	$g^D_{3,maxN}$
•				…				…	
•				…				…	
Gn		$g^N_{n,1}$	$g^N_{n,2}$	…	$g^N_{n,maxN}$	$g^D_{n,1}$	$g^D_{n,2}$	…	$g^D_{n,maxN}$

Table 1. Matrix of microarray expression dataset

2.2 Expression Sub-matrix m_S

We will form a candidate solution matrix C which consist of *d* randomly chosen non-recurring gene indexes from the gene dataset. A matrix m_s, $m_s \subset M$ has been obtained. For each index in candidate solution C, the corresponding entire row of M is selected and added in m_s. Here, $M_{index} = (1, 2, \dots, n)$ and *n* is the number of genes in the dataset. The expression sub-matrix m_s, candidate solution C and gene expression matrix M are shown in Fig. Here, we have taken an example where the no of rows in gene expression matrix M is 10. Each row is represented as *G1, G2, …, G10*. The candidate solution *C* has 3 elements. Each index i randomly chosen, $(i \in \{1, 2, 3, \dots, 10\})$ in C, represents to the row G_i in *M*. Each such row is added in m_s.

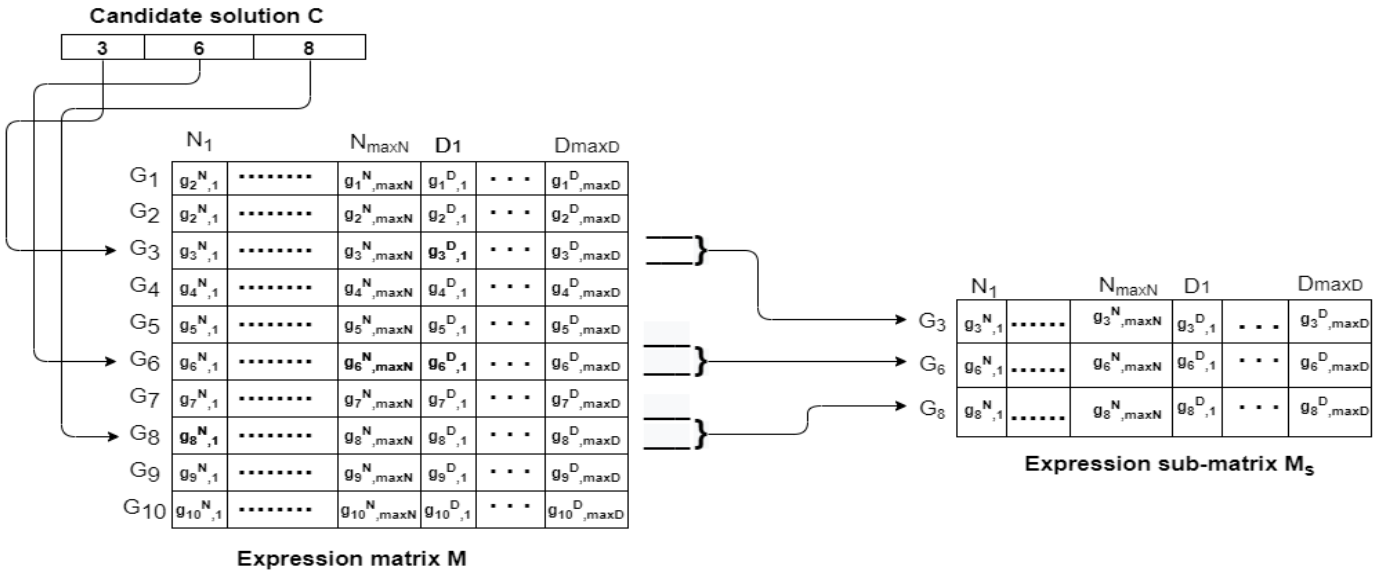


Fig 1.4 An example of retrieving m_s from C and M

2.3. Sample Classification of m_S

Using kNN, each particle in a sub-matrix m_s is classified. If the computed class of a particle in m_s has the similarity to its pre-existing class in *M*, the particular is represented as properly classified particle. The number of particles in *M* will have similarity to the number of particles in m_s. Here, classification accuracy (CA) is used to categorize.

$$CA = \frac{Number\ of\ properly\ classified\ samples\ in\ m_S}{Total\ number\ of\ samples\ in\ M} \times 100. \qquad (1)$$

3. Methodology

DE, a meta-heuristic algorithm is used for gene selection. The set of different critical genes is represented by length d which causes diseases in the candidate solution. From gene index matrix M_{index}, each element's gene index is chosen. The classifier kNN is used for calculating the fitness of candidate solutions. In section 3.1. how kNN is used to classify the gene samples is discussed. And in Section 3.2 the methodology of DE algorithms and its use to find the proper set of genes is discussed.

3.1. k Nearest Neighbor (kNN) Classification

In this portion, the Euclidian Distance is calculated between the t particles in test-set and each of (t −1) particles in train-set. The particles in train set are set in accordance to increasing order of their distance from the test particle. Top k (where k = 3) minimum particles are the nearest neighbors of the test particle [5]. In between the nearest neighbors, the majority of particles is computed (two out of three). If that particle has similar class to the test particle, then the test particle is properly classified. After calculation, the particle which is properly classified is given score 1 or else given score 0. Then after the process of t validation test, the sum of the individual scores for each and every particle gives the total properly classified particles in m_s.

Our main motive is to search for maximun number of properly classified particles in m_s for that reason the fitness score of candidate solution is evaluated by the sum of scores. The expression sub-matrix m_s is taken as input to score calculation. For cross-validation test, the current m^{th} column is kept in test set (Test), and the training set (Train) consists of all other columns without m^{th} column. For each particle n ∈ Train, the distance of n is computed from m. The equations for the particle m, n and distance dist_n_m is shown in Eq. (2), Eq. (3) and Eq. (4), respectively,

$$m = [\, g^{C_m}_{1,m\_ind},\, g^{C_m}_{2,m\_ind},\, g^{C_m}_{3,m\_ind}, \dots\, g^{C_m}_{d,m\_ind}] \,, \tag{2}$$

$$n = [\, g^{C_n}_{1,n\_ind},\, g^{C_n}_{2,n\_ind},\, g^{C_n}_{3,n\_ind}, \dots\, g^{C_n}_{d,n\_ind}], \tag{3}$$

$$Dist\_n\_m = \sqrt{\left(g^{C_n}_{1,n_{ind}} - g^{C_m}_{1,m_{ind}}\right)^2 + \left(g^{C_n}_{2,n_{ind}} - g^{C_m}_{2,m_{ind}}\right)^2 + \cdots + (g^{C_n}_{d,n_{ind}} - g^{C_m}_{d,m_{ind}})^2} \tag{4}$$

Here, C_m and C_n are the type of classes included in sample columns m and n,. The indexes of sample columns m and n in m_s are m_ind and n_ind.

3.2. Pseudocode of kNN Algorithm

```
kNN_fitness(Xs)
    fitnesss_score=0;
    arr={S1,S2,……,SmaxS,P1,P2,……..,PmaxP};
    for i=1 to n do
        test={i};
        train=arr-test;
        for j=1 to (n-1) do
            /*for each sample j in train*/
            calculate euclidian distance of j from I;
        end for
        sort samples solution in ascending order according to distance from i;
        get class Ti (Ti ∈ {normal,diseased}) of i;
        for j=1 to m do
            get Tj(Tj ∈ {normal,diseased}) of j;
        end for
        if Ti= =Tj
           then second_score=1;
        else
            second score=0;
    end for
    return fitness_score;
```

3.3. Differential Evolution Algorithm

Metaheuristic search algorithms are able to search with various and different levels of exploration and exploitation strengths with less computational effort. Meta means beyond and heuristic means to find, so it can able to search through a huge area. But the existing linear ,binary and other heuristic search algorithms can't explore huge areas so they can't find optimal solutions [8]. Because it is an optimized-based problem we need to find the optimal solution, that's why Metaheuristic algorithms are more useful.

To identify the critical genes we are using the Differential algorithm which is a category of Metaheuristic algorithm. It is a population-based Evolutionary algorithm. Due to the fact that Differential Evolution's search process is controlled by a few algorithm-specific parameters like the scaling factor and crossover rate, it is regarded as a reliable and straightforward method. Similar to other Evolutionary Algorithms, through mutation, crossover, and selection DE can produce new offspring solutions [7]. The search performance of the algorithm is more affected by mutation and crossover.

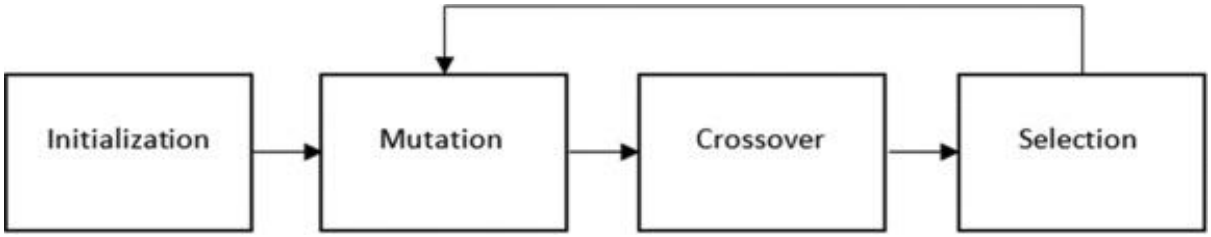


Fig 1.5-DE Algorithm major steps

Initialization

To clarify the algorithm steps we refer to the minimization problem of a benchmark function *f(Y),* where Y is an equation of any function.At first the search space is t*n,from that we randomy created initial search space of m*n dimension,here m*n defined by Z_{pop}.

Mutation

The biological definition of mutation is a sudden change in the characteristics of a chromosome gene. The mutation in the context of evolutionary computation is a random perturbation procedure applied to particular decision variables. [7].

At each iteration, three mutually distinct individuals Y_r , Y_s , and Y_t are extracted randomly from the population for each individual Y_i of the Z_{pop}.

The provisional offspring Y_{off} is generated by mutation as:

DE/rand/1: $Y'_{off} = Y_t + F(Y_r - Y_s)$ -------(5)

There are many strategies, but in this program, we used DE/rand/1, where rand means we have to take target vectors randomly and 1 means the number of vectors to target. where is a scale factor $F \in [0, +1]$ that controls the length of the exploration vector $(Y_r - Y_s)$ and determines the distance at which the offspring should be produced from point Yi. With $F \in [0, +1]$, it means that the scale factor value cannot be greater than 1 and less than 0 [8].

Crossover

In the crossover, a trial vector (offspring) is created when the components of the mutant and target vectors are crossed probabilistically. Due to the crossover process, the target solution can acquire the traits of the donor solution or mutant. [7]. Uniform crossover and exponential crossover are the most used crossover operators. The crossover is controlled by a crossover rate (*CR*) that has a value between [0,1].

When the provisional offspring has been generated by mutation, each gene of the individual Y_{off} is exchanged with the corresponding gene of Y_i with a uniform probability and the final offspring Y_{off} is generated as :

$Y_{off,j} = \{ Y_{i,j}$ if (CR >rand (0, 1))

$Y'_{off,j}$ otherwise

Selection

The resulting offspring Y_{off} is calculated and, according to the one-to-one spawning strategy, if ($f(Y_{off}) \leq f(Y_i)$) then we have to replace Y_i value with Y_{off} ; otherwise there will be no exchange.

3.4. Pseudocode of Differential Evolution Algorithm

```
while (iteration condition)
        for  i=1 to Z_pop
                compute f(Y_i)
        end-for
        for i=1 to  Z_pop
                //mutation//
                select Y_r ,Y_s and Y_t  indivudualy;
                Y'_off  = Y _t +F(Y _r -Y_s );
                //crossover//
                 Y_off =Y'_off ;
                for j = 1 to  n
                        x= random value(0, 1);
                        if (x < CR)
                   Y_off,j =Y_i,j ;
                      end-if
               end-for
              //selection//
              compute f(Y_off )
              If (f(Y_off ) <= f (Y_i ))
                 for j=1 to n
                   Y_i =Y_off
                 end-for
             end-if
         end-for
       replacements
end-while
```

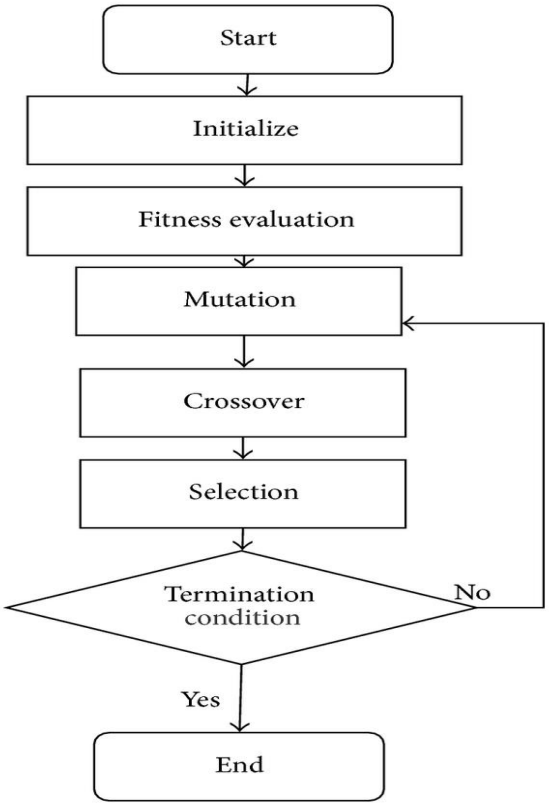


Fig 1.6 Differntial Evolution Algorithm Flowchart

4. Experimental Results

The machine that has been used to find the optimal or minimized value for the benchmark function has a windows 11 operating system,8GB RAM, GTX1050 graphics card, the code written in C++ language on VS code IDE software, Alpine1,Six-hump camel,Brain function code and Gene Selection code is written in Python on Jupyter Notebook, Below some graphs are generated through excel and some through Jupyter Notebook based on the optimized outputs.

4.1. Results of Benchmark Functions

For the optimization of the benchmark function, we used a dataset that has 100 rows and 5 columns, and from that, we initialized an initial search space of 10 rows and 5 columns. The values of search space are taken randomly from specific bounds.

The output table consists of benchmark functions output, the output came up after 50 runs and 200 iterations(each run). The table shows the mean, median, standard deviation, and, minimum and maximum values for each benchmark function. These are the minimized and optimal value from the search space.

4.1.1. List of Parameters :-

Name	Value	Functionality
F (Scaling Factor)	0.5	Scaling factor controls the length of the exploration vector.
CR (Crossover Probability)	0.7	It decides the rate of crossover and shows the number of features which inherited from the parent in the crossover.
Lb and ub(lower bound and upper bound)	-5.12,+5.12(varies according to function)	Lb and ub specify the range of the population.

4.1.2. Unimodal Benchmark Functions :-

Name	Function equiation	Range
Sphere	$\sum_{t=1}^{n} X_t^2$	[-5.12,+5.12]
Zakharov	$\sum_{t=1}^{n} X_t^2 + \left(\sum_{t=1}^{n} 0.5iX_t\right)^2 + \left(\sum_{t=1}^{n} 0.5iX_t\right)^4$	[-5,+10]
Dixon-price	$(X_1 - 1)2 + \sum_{t=2}^{n} i(2X_t^2 - X_{t-1})^2$	[-10,+10]
Rosenbrock	$\sum_{t=1}^{n-1} 100(X_t^2 - X_{t+1}^2)^2 + (X_t - 1)^2$	[-5,+10]

4.1.3. Multimodal Benchmark Functions :-

Name	Function equiation	Range
Ackly	$F7(X) = 20 + e - 20\exp\left(-0.2\sqrt{1/n\sum_{t=1}^{n} X_t^2}\right) - exp\left(\sqrt{1/n\sum_{t=1}^{n} cos(2\pi X_t)}\right)$	[-32,+32]
Levy	$F6(X) = sin^2(\pi X_t)\sum_{t=1}^{n-1}(Y_t - 1)^2[1+10sin^2(\pi Y_t + 1)] + (Y_t - 1)^2 [1+sin^2(2\pi Y_t)]\ where, Y_t = 1 + \frac{X_i - 1}{4}\ for i = 1,23,\ldots.n$	[-10,+10]

Rastrigin	$10n + \sum_{t=1}^{n}(X_t^2 - 10cos(2\pi X_t))$	[-5.12,+5.12]
Alpine1	$\sum_{i=1}^{D} \left\| x_i \sin(x_i) + 0.1x_i \right\|$	[-10,+10]

4.1.4. Fixed Dimensional Multimodal Benchmark Functions :-

Name	Function equiation	Range
Six-hump camel back	$4X_1^2 - 2.1X_1^4 + \frac{1}{3}X_1^6 + X_1X_2 - 4X_2^2 + 4X_2^4$	[-5,+5]
Brain	$(X_2 - \frac{5.1}{4\pi^2}X_1^2 + \frac{5}{\pi}X_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos X_1 + 10$	[-5,+5]

Most frequently used mutation strategies in DE.

"DE/rand/1" $Y'_{off} = Y_r + F(Y_s - Y_t)$

"DE/best/1" $Y'_{off} = Y_{best} + F(Y_r - Y_s)$

"DE/current-to-best/1" $Y'_{off} = Y_i + F(Y_{best} - Y_i) + F(Y_r - Y_s)$

"DE/rand/2" $Y'_{off} = Y_r + F(Y_s - Y_t) + F(Y_u - Y_v)$

"DE/best/2" $Y'_{off} = Y_{best} + F(Y_r - Y_s) + F(Y_t - Y_u)$

Below are the result of Benchmark function for all types of DE

4.1.5. Shpere Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
SphereRand1	7.68E-04	1.78E-13	0.005376	5.20E-17	3.84E-02
SphereBest1	2.72E-15	6.39E-20	1.47E-14	4.07E-24	9.98E-14
SphereCurrentBest1	6.82E-07	2.44E-15	4.75972E-06	9.71E-18	3.40E-05
SphereRand2	1.99E-11	7.44E-12	2.78917E-11	3.46E-13	1.38E-10
SphereBest2	2.72E-15	6.39E-20	1.47402E-14	4.07E-24	9.98E-14

4.1.6. Zakharov Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
ZakharovRand1	7.52E-02	2.26E-05	0.473895	2.90E-08	3.39E+00
ZakharovBest1	6.71E-05	3.50E-09	0.000313	4.83E-13	2.16E-03
ZakharovCurrentBest1	2.15E-03	4.26E-08	0.008326	1.04E-11	4.92E-02
ZakharovRand2	0.001964949	0.00041085	0.00447264	0.000005104	0.02061
ZakharovBest2	6.71E-05	3.50E-09	0.000313	4.83E-13	2.16E-03

4.1.7. Dixon-Price Function Output Tablefor All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
Dixon_PriceRand1	58.13286	51.99	39.5104	4.74	206.7
Dixon_PriceBest1	82.8108	69.11	72.86595	5.611	516.1
Dixon_PriceCurrentBest1	82.81082	69.11	72.86595	5.611	516.1
Dixon_PriceRand2	43.43485	46.475	26.10278	0.4604	141.5
Dixon_PriceBest2	82.81082	69.11	72.86595	5.611	516.1

4.1.8. Rosenbrock Function Output Tablefor All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
RosenbrockRand1	23.9876	12.75	28.59998	1.51	160
RosenbrockBest1	9.61734	2.66	16.55876	0.332	82.1
RosenbrockCurrentBest1	6.41142	2.835	7.927052	0.245	32.5
RosenbrockRand2	48.1452	21.6	58.22984	2.64	225
RosenbrockBest2	9.61734	2.66	16.55876	0.332	82.1

4.1.9. Ackly Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
AcklyRand1	2.732584	2.801	1.821831	0.001944	6.456
AcklyBest1	1.63723121	1.519	1.72021582	0.00001725	6.418
AcklyCurrentBest1	1.862804	1.7445	1.757328	0.000162	6.039
AcklyRand2	4.381412	4.4655	1.92811	0.1924	9.209
AcklyBest2	1.63723121	1.519	1.72021582	0.00001725	6.418

4.1.10. Restrigin Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
RastriginRand1	0.987072	0.996	0.858294	0	3.786
RastriginBest1	1.71E+00	1.49E+00	1.712361	0	6.97E+00
RastriginCurrentBest1	0.756585	0.5422	0.977738	0	3.984
RastriginRand2	0.19923426	9.5345E-06	0.48792438	0	1.992
RastriginBest2	1.71E+00	1.49E+00	1.712361	0	6.97E+00

4.1.11. Levy Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
LevyRand1	1.62E-02	1.03E-06	0.073769	3.34E-09	5.11E-01
LevyBest1	2.77E-02	4.84E-10	0.091302	0	4.56E-01
LevyCurrentBest1	3.60E-03	2.04E-07	0.017492	3.71E-11	8.93E-02
LevyRand2	8.45E-05	8.57E-06	0.000215	5.48E-08	1.10E-03
LevyBest2	2.77E-02	4.84E-10	0.091302	0	4.56E-01

4.1.12. Alpine1 Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
Alpine1Rand1	4.77E-03	1.56E-05	0.014909	7.91E-09	9.17E-02
Alipne1Best1	4.10E-03	1.91E-07	0.025965	1.64E-10	1.85E-01
Alipne1CurrentBest1	1.05E-03	4.05E-06	0.007037	1.18E-08	5.03E-02
Alipne1Rand2	0.00054566	0.00012	0.00236374	0.00000198	0.017
Alipne1Best2	1.05E-03	4.05E-06	0.007037	1.64E-10	5.03E-02

4.1.13. Six-hump camel back Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
Six-hump camel back Rand1	-1.03116	-1.0316	0.003009	-1.031	-1.0101
Six-hump camel back Best1	-1.0316	-1.0316	4.44089E-16	-1.0316	-1.0316
Six-hump camel back Current Best1	-1.0316	-1.0316	4.44089E-16	-1.0316	-1.0316
Six-hump camel back Rand2	-1.0316	-1.0316	4.44089E-16	-1.0316	-1.0316
Six-hump camel back Best2	-1.0316	-1.0316	4.44089E-16	-1.0316	-1.0316

4.1.14. Brain Function Output Table for All Types of DE:-

Benchmark Functions	Mean	Median	Standard deviation	Minimum	Maximum
Brain Rand1	0.403339	0.39809	0.036744	0.39809	0.66055
Brain Best1	0.39809	0.39809	1.11022E-16	0.39809	0.39809
Brain CurrentBest1	0.39809	0.39809	1.11022E-16	0.39809	0.39809
Brain Rand2	0.559392	0.39809	1.129115	0.39809	8.4632
Brain Best2	0.39809	0.39809	1.11022E-16	0.39809	0.39809

4.1.6. Unimodal Output Table of All Minimum Values:-

Benchmark Functions	Minimum	Known Minimum
Sphere	4.07E-24	1.83E-14
Zakharov	4.83E-13	2.84E-09
Dixon-Price	0.4604	5.72E-4
Rosenbrock	0.245	0

4.1.7. Multimodal Output Table of All Minimum Values:-

Benchmark Functions	Minimum	Known Minimum
Ackly	0.00001725	1.01E-06
Rastrigin	0.00E+00	2.7E-12
Levy	3.71E-11	6.46E-06
Alpine1Rand1	1.64E-10	0

4.1.8. Fixed Dimension Multimodal Output Table of All Minimum Values:-

Benchmark Functions	Minimum	Known Minimum
Six-hump camel back	-1.031	-1.0316
Brain	0.397	0.398

4.1.9 Convergency Analysis of Differential Evolution Algorithm(DE)

Sphere

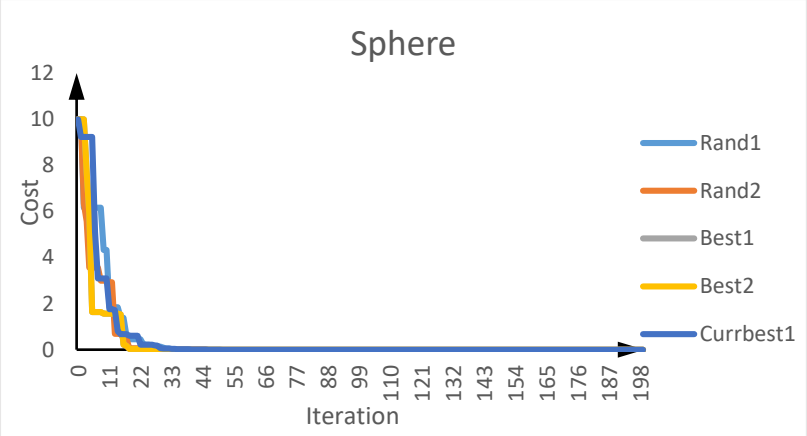


Fig 1.7- Convergence graph for Sphere function

Zakharov

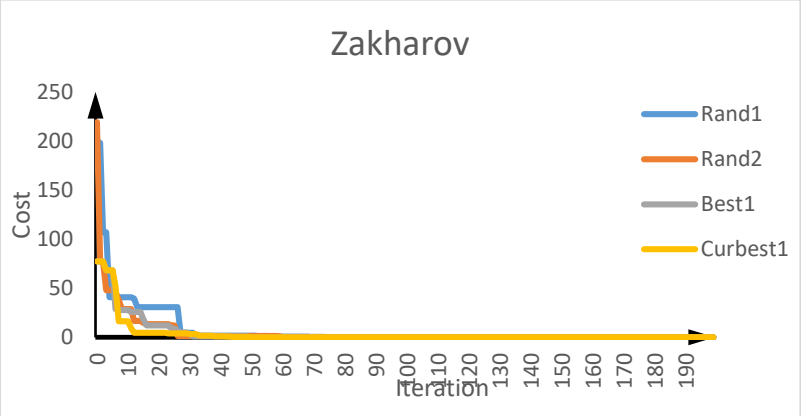


Fig 1.8- Convergence graph for Zakharov function

Dixon_price

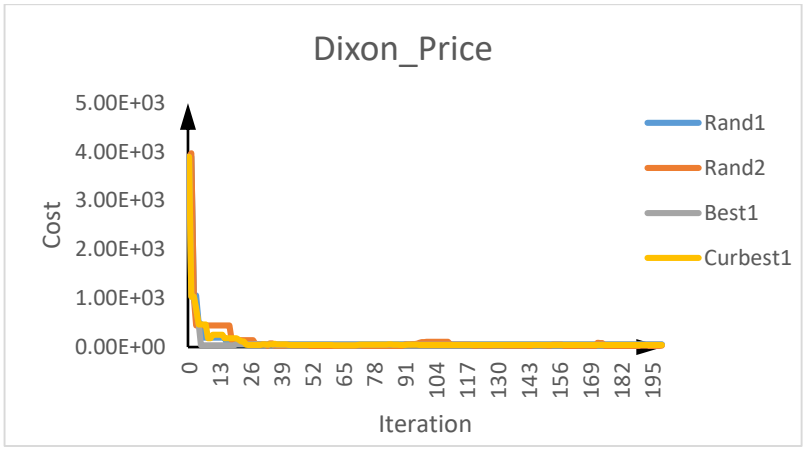


Fig 1.9- Convergence graph for Dixon_Price function

Rosenbrock

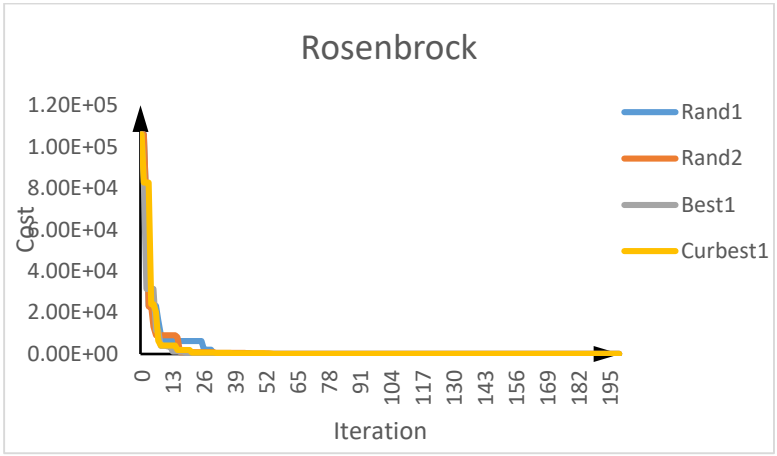


Fig 1.10- Convergence graph for Rosenbrock function

Ackly

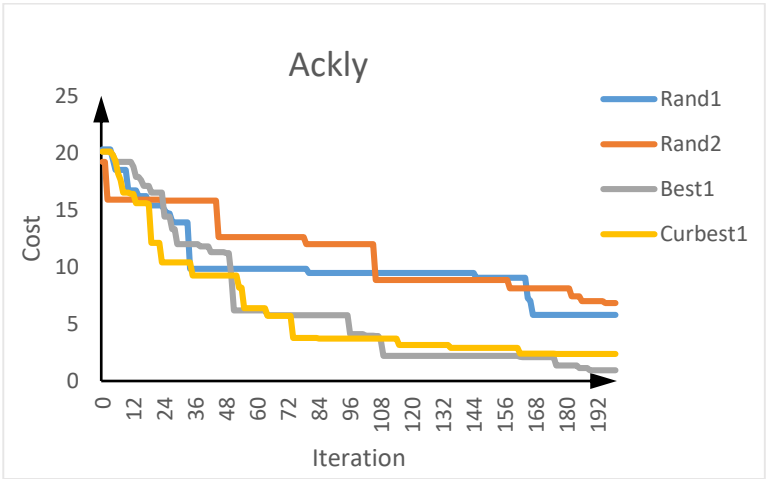


Fig 1.11- Convergence graph for Ackly function

Levy

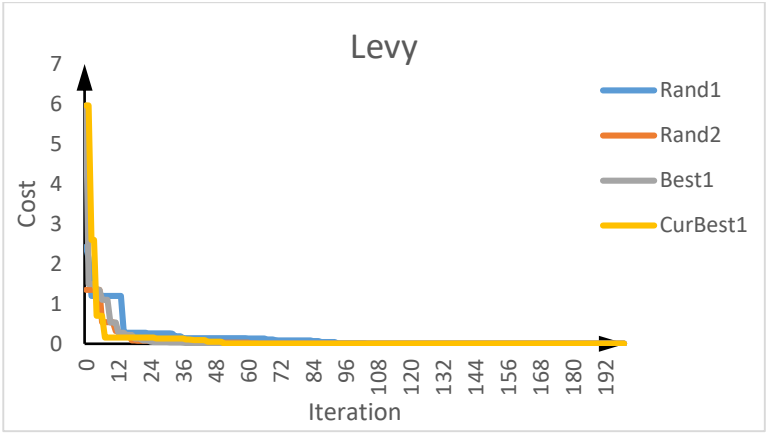


Fig 1.12- Convergence graph for Levy function

Rastrigin

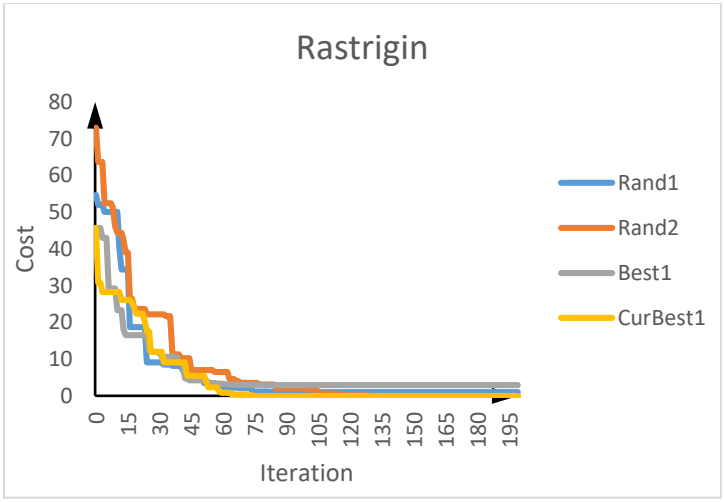


Fig 1.13- Convergence graph for Rastrigin function

Alpine1

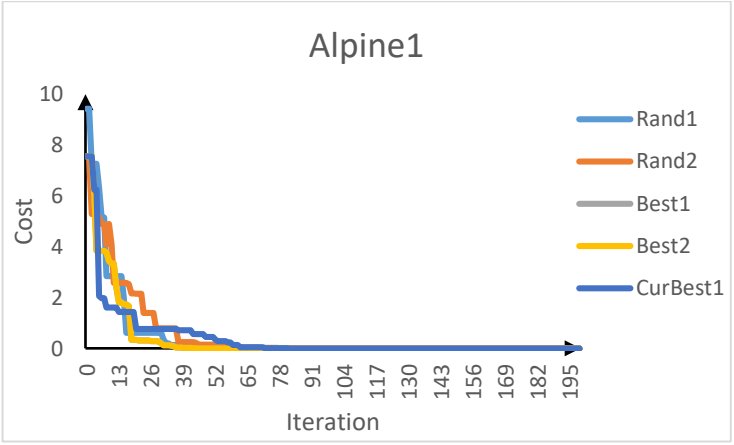


Fig 1.14- Convergence graph for Alpine1 function

Six-hump camel back

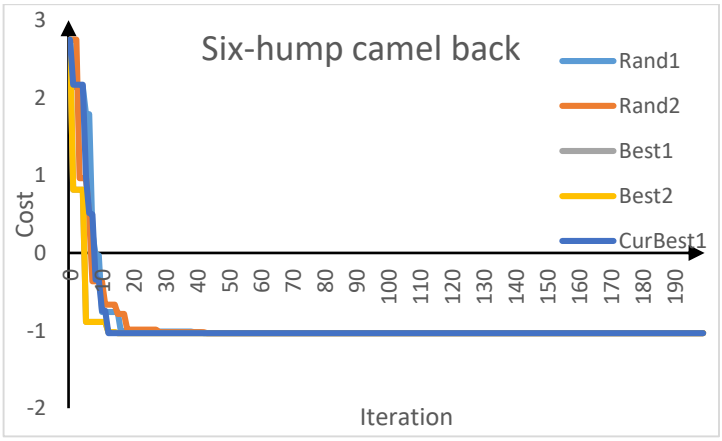


Fig 1.15- Convergence graph for Six-hump camel back function

Brain

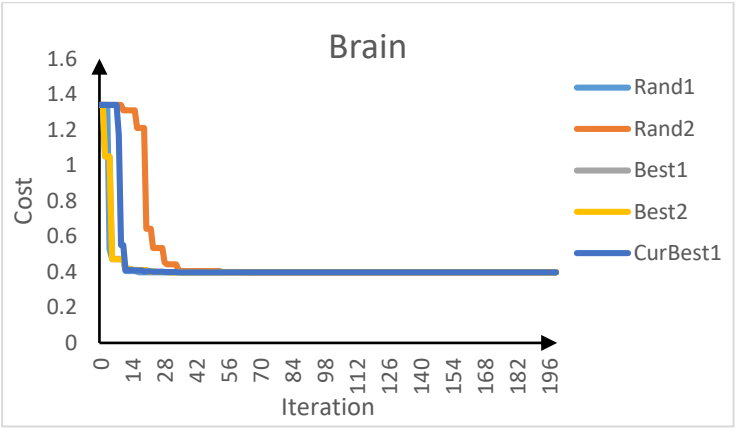


Fig 1.16- Convergence graph for Brain function

4.2 Result on Critical Genes Identification

4.2.1. Data and Experimental Setting

Two gene expression dataset with accession number GSE6613_details and GSE6613_dataset have been used in this research.Each of the dataset has been produced through an microarray experiment.GSE6613 contains 22284 number of gene ids and 70 samples(disease and normal).

Each elemnet of gene expression dataset contains floating point values.So,the dataset has been normalized.The normalization method works as,the maximum and minimum sample values of each genes has been calculated first using the Eq (6)

$$\text{Normalized}(g_{i,j}) = \frac{g_{i,j} - G_i^{MIN}}{G_i^{MAX} - G_i^{MIN}} \qquad (6)$$

4.2.2. Evaluation Metrics

Algorithm	Parameters
DE-KNN	Iteration=1000, Run=10, F(scaling Factor)=0.5 CR(Crossover Rate)=0.7, No of Solution=10, No of candidates(Dimension)=30

For the sake of fairness the number of iterations is set to 1000, solution set as 10 and the dimension of each sample is set to 30. Further to reduce the influence of random number on each algorithm, 10 independent runs are taken. The experimental results of fitness score(Global Best) have been presented in Table1.

4.2.3 Best Result Obtained from Dataset of PD

Dataset	Algorithm	Avg. Fitness	Best Fitness	Clasification Accuracy on Best fitness(%)
GSE6613[70]	DE-kNN	52.216	60	86.2

4.2.4 Output from Best Fitness

Mean	52.216
Median	52
Minimum	50
Maximum	60
Standard Deviation	0.6952294585243062

Figure 1.17 shows fitness score convergence graph of DE based on dataset GSE6613.Here the number of iterations is shown in x-axis(1000) and the fitness score is shown in y-axis .

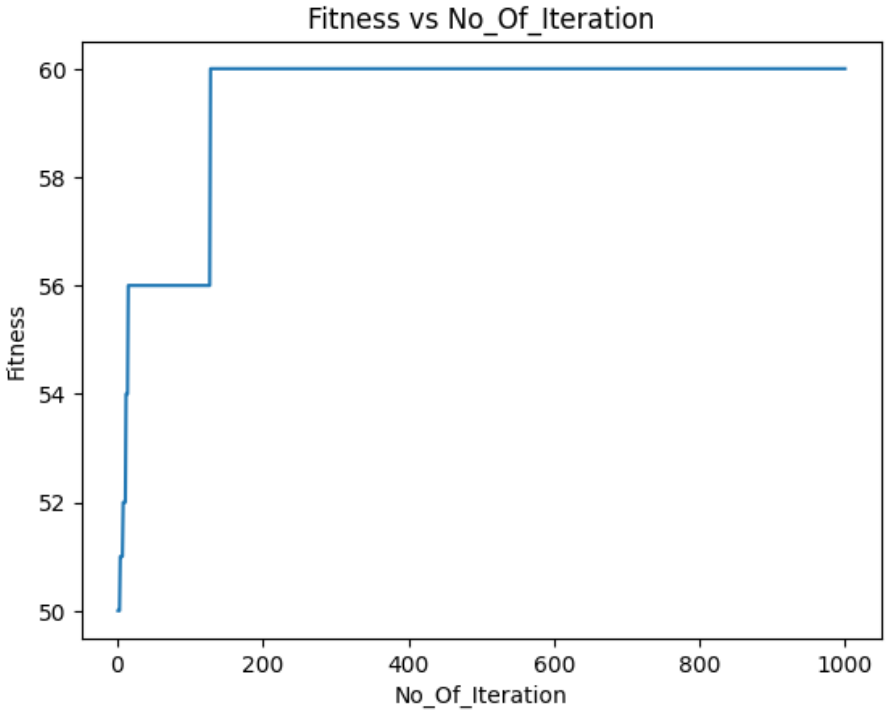


Fig 1.17- Fitness score Divergence graph of the execution based on GSE6613

4.2.3. Ditected Critical Gene Id

Sample Number	Gene id
12493	213109_at
6512	206985_at
3790	204262_s_at
7291	207769_s_at
11448	212061_at
8566	209071_s_at
9874	210395_x_at
18201	218836_at
14770	215395_x_at
6522	**206995_x_at***
11280	211890_x_at
15784	216412_x_at
9377	209890_at
8630	**209135_at***
11430	**212043_at***
15645	216272_x_at
4892	205364_at
5801	206274_s_at
15489	216116_at
13247	213866_at
15375	216002_at
12478	**213094_at***
9883	205364_at
3905	204377_s_at
6742	**207216_at***
8926	209432_s_at
16697	217331_at
9746	210265_x_at
8382	208886_at
9412	209925_at

(*)gene ids(matched) are already been ditected by other reseachers

5. Conclusion and Future Work

Overall, the different variants of DE have different strengths and weaknesses and can be used in various optimization problems depending on their requirements. Researchers continue to explore new variants of DE and hybrid algorithms to further improve its performance and widen its range of applications.

The identification of disease-critical genes causing Parkinson's disease is a complex problem that requires the analysis of large amounts of genomic data. Differential Evolution Algorithm(DE) has shown promise in identifying disease-critical genes in Parkinson's disease, and there is still a lot of scope for future research in this area.

One potential future direction for DE research in DE is to incorporate more complex and varied data sources into the optimization process. For example, DE can be applied to analyze gene expression data, protein-protein interaction data, and epigenetic data, among other sources. By combining these data sources and leveraging the strengths of DE, researchers can gain a deeper understanding of the molecular mechanisms underlying DEand identify new potential drug targets.

Another area of future research is the development of new DE-based models that can handle multiple objectives and constraints simultaneously. For example, the identification of disease-critical genes may involve optimizing multiple objectives such as minimizing false positives, maximizing sensitivity, and minimizing computational time. DE can be extended to handle multiple objectives and constraints, allowing researchers to explore the trade-offs between different objectives and identify the best solutions.

In conclusion, there is still a lot of scope for future research in the identification of disease-critical genes causing Parkinson's disease using Differential Evolution Algorithm(DE). By incorporating more complex data sources, developing multi-objective models, creating hybrid algorithms, and applying DE to other disease areas, researchers can further enhance the performance and widen the range of applications of this algorithm.

6. References

[1] Stekel, D. (2003). *Microarray bioinformatics*. Cambridge University Press.

[2] Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J.M. and Herrera, F.2014. A review of microarray datasets and applied feature selection methods. *Information sciences*, *282*, pp.111-135.

[3] Davie, C. A. (2008). A review of Parkinson's disease. *British medical bulletin*, *86*(1), 109-127.

[4] Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (Eds.). (2013). *Metaheuristic applications in structures and infrastructures*. Newnes.

[5] Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, *26*(3), 392-398.

[6] Miller, D. B., & O'Callaghan, J. P. (2015). Biomarkers of Parkinson's disease: present and future. *Metabolism*, *64*(3), S40-S46.

[7] Ahmad, M. F., Isa, N. A. M., Lim, W. H., & Ang, K. M. (2022). Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*, *61*(5), 3831-3872.

[8] Das, Swagatam, and Ponnuthurai Nagaratnam Suganthan. "Differential evolution: A survey of the state-of-the-art." *IEEE transactions on evolutionary computation* 15, no. 1 (2010): 4-31.

[9] Oti, M., & Brunner, H. G. (2007). The modular nature of genetic diseases. *Clinical genetics*, *71*(1), 1-11.

[10] https://www.researchgate.net/figure/Example-of-gene-expression-data-of-Alzheimers-Disease_fig2_345480824

[11] https://www.researchgate.net/figure/Outline-of-gene-expression-analysis-by-DNA-microarray-technology-RNA-is-extracted-from_fig2_228418971

[12] https://www.indushealthplus.com/genetic-dna-testing/most-common-genetic-disorders.html

[13] https://en.wikipedia.org/wiki/Parkinson%27s_disease

[14] Das, S. and Suganthan, P.N., 2010. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, *15*(1), pp.4-31.

[15] Pant, M., Zaheer, H., Garcia-Hernandez, L. and Abraham, A., 2020. Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, *90*,pp.103479.

[16] Neri, F. and Tirronen, V., 2010. Recent advances in differential evolution: a survey and experimental analysis. *Artificial intelligence review*, *33*, pp.61-106.

[17] Das, S., Mullick, S.S. and Suganthan, P.N., 2016. Recent advances in differential evolution–an updated survey. *Swarm and evolutionary computation*, *27*, pp.1-30.

[18] Jebaraj, L., Venkatesan, C., Soubache, I. and Rajan, C.C.A., 2017. Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: A review. *Renewable and Sustainable Energy Reviews*, *77*, pp.1206-1220.

[19] Opara, K.R. and Arabas, J., 2019. Differential Evolution: A survey of theoretical analyses. *Swarm and evolutionary computation*, *44*, pp.546-558.

[20] Javaid, N., 2019. Differential evolution: An updated survey. In *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018)* (pp. 681-691). Springer International Publishing.

[21] Biswas, S., Dutta, S., & Acharyya, S. (2019). Identification of disease critical genes using collective meta-heuristic approaches: an application to preeclampsia. *Interdisciplinary Sciences: Computational Life Sciences*, *11*(3), 444-459.

[22] Magoulas, G. D., & Prentza, A. (2001). Machine learning in medical applications. *Machine Learning and Its Applications: advanced lectures*, 300-307.